

Міністерство освіти і науки України
Національний університет «Львівська політехніка»



Звіт

до лабораторної роботи №4

З дисципліни: «Кросплатформенні засоби програмування»

На тему: «Виключення»

Варіант 8

Виконав:

Киянець А.М.

Ст. групи КІ- 306

Прийняв:

Іванов Ю.С.

Львів 2023

Мета

Оволодіти навиками використання механізму виключень при написанні програм мовою Java.

Контрольні питання

1. Дайте визначення терміну «виключення».

Відповідь: Виключення – це механізм мови Java, що забезпечує негайну передачу керування блоку коду опрацювання критичних помилок при їх виникненні уникаючи процесу розкручування стеку.

2. У яких ситуаціях використання виключень є виправданим?

Відповідь: помилках введення, збоях обладнання, помилках, що пов'язані з фізичними обмеженнями комп'ютерної системи та помилках програмування

3. Яка ієрархія виключень використовується у мові Java?

Відповідь: Всі виключення в мові Java поділяються на контрольовані і неконтрольовані та спадкуються від суперкласу Throwable. Безпосередньо від цього суперкласу спадкуються 2 класи Error і Exception.

4. Як створити власний клас виключень?

Відповідь: заекстендити Error або Exception(або дочірні).

5. Який синтаксис оголошення методів, що можуть генерувати виключення?

Відповідь: [access modifier] [return type] [name]([parameters]) throws Exception

```
{  
...  
}
```

6. Які виключення слід вказувати у заголовках методів і коли?

Відповідь: Оголошувати слід лише всі контрольовані виключення. Якщо цього не зробити, то компілятор видасть повідомлення про помилку. Якщо метод оголошує, що він може генерувати виключення певного класу, то він може також генерувати виключення і його підкласів.

7. Як згенерувати контрольоване виключення?

Відповідь:

1. throw new Exception();
2. Exception ex = new Exception(); throw ex;

8. Розкрийте призначення та особливості роботи блоку try.

Відповідь: Блок виконання коду.

9. Розкрийте призначення та особливості роботи блоку catch.

Відповідь: Блок перехоплення виключення.

10. Розкрийте призначення та особливості роботи блоку finally.

Відповідь: Блок гарантованого виконання коду після try-catch блоку.

Завдання

23. $y = \text{ctg}(8x)/x$

1. Створити клас, що реалізує метод обчислення виразу заданого варіантом. Написати на мові Java та налагодити програму-драйвер для розробленого класу. Результат обчислень записати у файл. При написанні програми застосувати механізм виключень для виправлення помилкових ситуацій, що можуть виникнути в процесі виконання програми. Програма має розміщуватися в пакеті Група.Прізвище.Lab5 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.

2. Автоматично згенерувати документацію до розробленого пакету.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання.

Код програми

EquationsApp.java

```
package Ki306.Kyianets.Lab4;
import java.util.Scanner;
import static java.lang.System.out;
import java.io.*;

public class EquationsApp {
    public static void main(String[] args) {
        try {
            out.print("Enter file name: ");
            Scanner in = new Scanner(System.in);
            String fName = in.nextLine() + ".txt";
            PrintWriter fout = new PrintWriter(new File(fName));
            try {
                out.print("Enter X: ");
                double x = in.nextDouble();
                double result = EquationCalculator.calculate(x);
                fout.print(result);
            } catch (CalcException ex) {
                out.print(ex.getMessage());
            }
        }
    }
}
```

```

        } finally {
            fout.flush();
            fout.close();
        }
    } catch (FileNotFoundException ex) {
        out.print("Exception reason: Perhaps wrong file path");
    }
}

class CalcException extends ArithmeticException {
    public CalcException() {}

    public CalcException(String message) {
        super(message);
    }
}
}
}

```

EquationCalculator.java

```

package Ki306.Kyianets.Lab4;

public class EquationCalculator {
    public static double calculate(double x) {
        double y;
        try {
            y = Math.sin(x) / Math.sin(2 * x - 4);
        } catch (ArithmeticException ex) {
            // Обробити помилки обчислень
            throw new ArithmeticException("Exception reason: " +
ex.getMessage());
        }
        return y;
    }
}

```

Скріншоти роботи програми після захисту

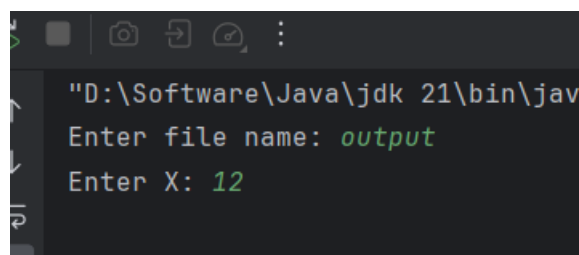
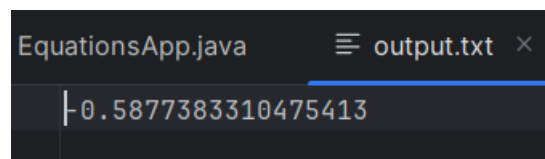
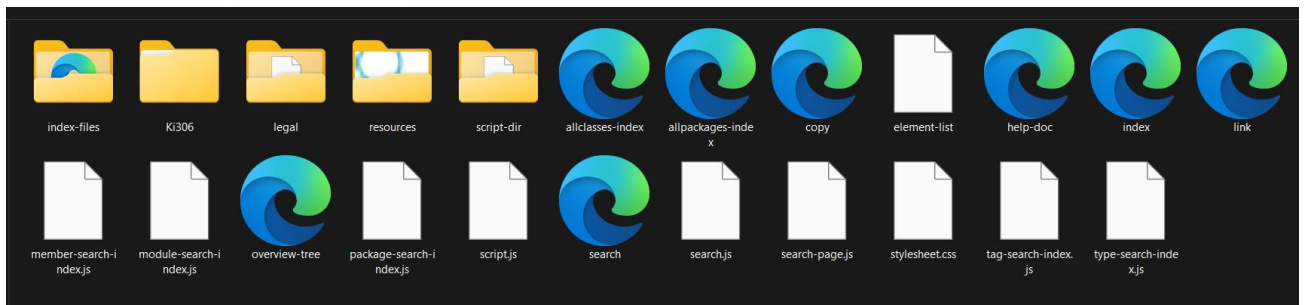


Рис. 1. Результат роботи програми.



Сформована Java документація



Висновок

Я оволодів навиками використання механізму виключень при написанні програм мовою Java.